Microchip Debugger (MDB) User's Guide

© 2012-2013 Microchip Technology Inc.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2012-2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 978-1-62077-171-6

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

© 2012-2013 Microchip Technology Inc.

Table of Contents

Preface 5
How to Use MDB 9
Index 19
Worldwide Sales and Service 21

NOTES:

© 2012-2013 Microchip Technology Inc.

Preface

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site (www.microchip.com) to obtain the latest documentation available.

Documents are identified with a "DS" number. This number is located on the bottom of each page, in front of the page number. The numbering convention for the DS number is "DSXXXXXXXXA", where "XXXXXXXX" is the document number and "A" is the revision level of the document.

For the most up-to-date information on development tools, see the MPLAB® X IDE help. Select the Help menu, and then Topics to open a list of available help files.

INTRODUCTION

This chapter contains general information that will be helpful to know before using the Microchip Debugger (MDB). Items that are discussed include:

- Conventions Used in This Guide
- Recommended Reading
- Revision History

CONVENTIONS USED IN THIS GUIDE

The following conventions may appear in this documentation:

TABLE 1: DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial font:		
Italic	Referenced books	*MPLAB® IDE User's Guide*
	Emphasized text	...is the *only* compiler...
Initial caps	A window	the Output window
	A dialog	the Settings dialog
	A menu selection	select Enable Programmer
Quotes	A field name in a window or dialog	"Save project before build"
Underlined, italic text with right angle bracket	A menu path	*File>Save*
Bold	A dialog button	Click **OK**
	A tab	Click the **Power** tab
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier font:		

TABLE 1: DOCUMENTATION CONVENTIONS (CONTINUED)

Description	Represents	Examples
Plain	Sample source code	`#define START`
	Filenames	`autoexec.bat`
	File paths	`c:\mcc18\h`
	Keywords	`_asm, _endasm, static`
	Command-line options	`-Opa+, -Opa-`
	Bit values	`0, 1`
	Constants	`0xFF, 'A'`
Italic	A variable argument	*`file`*`.o`, where *`file`* can be any valid filename
Square brackets [ ]	Optional arguments	`mpasmwin [options]` *`file`* `[options]`
Curly brackets and pipe character: { | }	Choice of mutually exclusive arguments; an OR selection	`errorlevel {0|1}`
Ellipses...	Replaces repeated text	`var_name [, var_name...]`
	Represents code supplied by user	`void main (void) { ... }`

RECOMMENDED READING

This document describes how to use the Microchip Debugger (MDB). Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

Development Tools Design Advisory (DS51764)

A small document on guidelines and implementation considerations to ensure proper interfacing to the various development tools.

Processor Extension Pak and Header Specification (DS51292)

This booklet describes how to install and use Processor Extension Paks (PEPs) and related debug headers to better debug selected devices without the loss of pins or resources. See also the PEP and Header online help file.

Transition Socket Specification (DS51194)

Consult this document for information on transition sockets available for use with headers.

© 2012-2013 Microchip Technology Inc.

REVISION HISTORY

Revision A (November 2012) - initial release of this document.

Revision B (April 2013)

- added revision history section
- added note in Invoking the MDB section
- added Tool Property Name Options for the Set command
- added `-p` option
- added note on running multiple tools
- removed example of using commands to debug a project
- added section on creating a printable log file

NOTES:

© 2012-2013 Microchip Technology Inc.

How to Use MDB

INTRODUCTION

MDB is the Microchip Debugger. It facilitates programming and debugging devices through the Command Prompt interface, instead of using the Microchip MPLAB® X IDE (Integrated Development Environment). MDB was designed for engineers who prefer to use the Command Prompt.

GETTING STARTED

Install MPLAB X IDE. See the Microchip web site (www.microchip.com) to download the latest version. The MDB is automatically installed with the MPLAB X IDE.

Generate a cof/elf file for debugging. (If simply programming a device, a hex file is sufficient.) The project can be built with MPLAB X IDE or using third-party compilers, as long as a cof/elf file is generated. The cof/elf file is a linked executable file that contains symbolic debugging information.

INVOKING THE MDB

Use the Command Prompt to invoke MDB.

On Windows 7, the Command Prompt must be opened in Administrator mode: *Start>All Programs>Accessories>Command Prompt*, right click and select "Run as Administrator." This opens the Administrator: Command Prompt.

The path to the MDB may be vary, depending on where the MPLAB X IDE was installed, and which operating system is installed.

> **Note:** The mdb.bat and mdb.sh scripts do not need to be run from the directory where they were installed. If the directory where these scripts are installed is added to the system path, then mdb.bat and msb.sh may run from any directory.

In Windows 32 bit:

```
c:\>cd "c:\Program Files\Microchip\MPLABX\mplab_ide\bin"
c:\Program Files\Microchip\MPLABX\mplab_ide\bin>mdb.bat
```

In Windows 64 bit:

```
c:\>cd "c:\Program Files (x86)\Microchip\MPLABX\mplab_ide\bin"
c:\Program Files (x86)\Microchip\MPLABX\mplab_ide\bin>mdb.bat
```

In Linux:

```
$ cd /opt/microchip/mplabx/mplab_ide/bin
$ ./mdb.sh
```

In Mac OSX:

```
$ cd
/Applications/microchip/mplabx/mplab_ide.app/Contents/Resources/mplab_
ide/bin
$ ./mdb.sh
```

© 2012-2013 Microchip Technology Inc.

DEBUGGING METHODS

You can run a test using either of these methods:

- Entering Commands Method
- Running a Command File Method

Entering commands is the preferred method to run a test with MDB. It allows you to interact with the target application as it executes in simulation or on actual hardware. The result of each command is displayed one at a time, so that mistakes are more easily understood and corrected. See "Entering Commands Method".

Running a command file method cannot be used after invoking the MDB. It is included as a parameter in the command line when invoking the MDB. See "Running a Command File Method".

ENTERING COMMANDS METHOD

> **Note:** Although the MPLAB X IDE can run multiple tools simultaneously, the MDB will not be able to identify each tool.

Entering commands is a step-by-step method to run a test with MDB. The following sections describe:

- Classes of Commands
- List of Commands
- Programming a Device
- Debugging a Device

Classes of Commands

Once the MDB is running, you can start entering commands. Please note that the MDB commands are not case-sensitive. Type `help` for a list of classes of commands in MDB.

TABLE 1-1: MDB CLASSES OF COMMANDS

breakpoints	Making program stop at certain points
data	Examining/changing data
deviceandtool	Selecting debug tool and device
others	Miscellaneous commands
programming	Programming device and its relative functions
running	Running the program
stack	Examining stack

© 2012-2013 Microchip Technology Inc.

List of Commands

For a list of all commands within a particular class, type `help` followed by the class name. See the following tables for information about each list of commands.

For documentation on a particular command, type `help` followed by the command name.

TABLE 1-2: BREAKPOINTS - LIST OF COMMANDS

Break	Set a breakpoint at the specified source line number:
	`break filename:linenumber [passCount]`
	
	Set a breakpoint at an absolute address:
	`break *address [passCount]`
	
	• address - The address of the data memory to be watch. Use command `'print /a'` to get a variable address.
	• passCount - The parameter is optional. The number of times the breakon condition is met before the program breaks.
	
	Example:
	`break newmain.c:16`
	
	MDB assigns a breakpoint number and returns:
	`Breakpoint 0 at 0x9d0000cc: file newmain.c, line 16.`

Watch	Set a data breakpoint at the specified memory address:
	`Watch address breakonType[:value] [passCount]`
	
	• address - The address of the data memory to be watched. Use command `'print /a'` to get a variable address.
	• BreakonType:
	`R` -- Read.
	`W` -- Write.
	`RW` -- Read or Write.
	• value - The parameter is optional. If it is specified, the program will break only when the value held in the data memory matches the specified value matches the specified value.
	• passCount - The parameter is optional. The number of times the breakon condition is met before the program breaks.
	
	Examples:
	`watch 0xfffff W`
	
	MDB assigns a watchpoint number and returns:
	`Watchpoint 1`

Delete	Delete a breakpoint - If no argument is specified, delete all breakpoints. You can abbreviate this command as `d`.
	`Delete [breakpoint number]`
	
	• breakpoint number - Generated by MDB for break and/or watch command.
	
	Example:
	`Delete 1`

Halt	Stop the debugged program.

TABLE 1-3: DATA - LIST OF COMMANDS

Print	Print a variable with optional formatting. `print [/f] [/datasize:value] variable.` • `/f` - Optional format letter. The format letters supported are: `x` - Print as integer in signed hexadecimal. `d` - Print as integer in signed decimal. `a` - Print the address of a symbol. • `/datasize:value` - Optional data size. Variable in assembly code might not have data size information. User can specify the data size if the .cof or .elf file does not have the size information. The values supported are: `1` - The data size is 1 byte. `2` - The data size is 2 bytes. `4` - The data size is 4 byte.

Stim	Specify a simulator SCL stimulus file to use. Loads the specified SCL stimulus file into the simulator, or if no path to the file is specified, it clears a loaded file. (Note, if the path or filename has spaces in it, you must use the quotation marks as shown below. If there are no spaces in the path of filename, the quotation marks are not needed.)
	`Stim "[path to file]"`
	`Stim`

x	Examine memory. You can use the command x (for examine) to examine memory in any of several formats, independently of your program's data types. `x [/tnfu] [addr]` • t - the type of memory. Each time you specify a memory type with x, that type becomes the default memory the next time you use x.The type of memory is any of the following: r File Registers (RAM) memory. This is the initial default. p Program (flash) memory. e EE Data memory. • n - the repeat count. The repeat count is a decimal integer; the default is 1. It specifies how much memory (counting by units u) to display. • f - the display format. The display format is one of the formats used by print (x, d, o, f, s), and in addition "i" (for machine instructions). The default is 'x' (hexadecimal) initially. The default changes each time you use x. • u - the unit size. Each time you specify a unit size with x, that size becomes the default unit the next time you use x. (For the 's' and 'i' formats, the unit size is ignored and is normally not written.) The unit size is any of: b Bytes. h Halfwords (two bytes). w Words (four bytes). This is the initial default. • addr - starting display address. addr is the address where you want MDB to begin displaying memory. The expression need not have a pointer value (though it may); it is always interpreted as an integer address of a byte of memory.The default for addr is usually just after the last address examined, but several other commands also set the default address: info breakpoints (to the address of the last breakpoint listed), info line (to the starting address of a line), and print (if you use it to display a value from memory).

© 2012-2013 Microchip Technology Inc.

TABLE 1-4: DEVICEANDTOOL - LIST OF COMMANDS

Device	Set the name of the target device.
	`Device devicename`
	
	Example:
	`Device PIC32MX795F512L`

Hwtool	Set the debug tool.
	`Hardware toolname`
	
	Following are the supported tool names (not case-sensitive):
	• ICD3 - MPLAB ICD 3 In-Circuit Debugger
	• RealICE - MPLAB REAL ICE In-Circuit Emulator
	• PICkit3 - PICkit 3 In-Circuit Debugger
	• SIM - Simulator
	• PM3 - MPLAB PM3 Programmer
	• LicensedDebugger - third party debugger
	• LicensedProgrammer - third party programmer
	
	Note: By default, when a hardware tool is selected, it is loaded as a debugger. This means that it always programs the device and adds the necessary debug requirements to enable the image to be debugged.
	To use a tool for programming only, use the -p option when setting the hardware tool.
	
	Set the tool for programming only. A space must precede the `-p` option.
	`Hardware toolname -p`
	
	Example:
	`Hwtool ICD3 -p`

TABLE 1-5: OTHERS - LIST OF COMMANDS

Help	Print list of commands.
Quit	Quit - exits the debugger.

Set	Set command. The tool property name and value are from the project properties that are selected when creating the project in MPLAB X IDE.
	`Set tool-property-name value`
	
	Example:
	`Set programoptions.eraseb4program true`
	
	Refer to Table 1-6 for other tool properties that can used with the `Set` command.

Sleep	Makes the current script processor sleep until specified milliseconds have elapsed.
	`Sleep milliseconds`
	
	Example:
	`Sleep 10`

Wait	Wait command makes the current script processor wait until the debugger halts before processing the next command.
	`Wait`
	
	Wait Milliseconds makes the processor process the next command if the debugger does not halt and milliseconds have elapsed.
	`Wait [milliseconds]`

TABLE 1-6: TOOL-PROPERTY-NAME OPTIONS USED WITH SET COMMAND

Tool Property Name	Value
AutoSelectMemRanges	auto or none
Determines whether the debugger will automatically select the areas of memory and program memory ranges to program. If set to auto the debugger will automatically select the memory and ranges. None means the memories and ranges will be determined by the memories properties below.	
memories.programmemory	true or false
If true, the program memory will be programmed, if false, it will not.	
memories.eeprom	true or false
If true, the EEPROM memory will be programmed, if false, it will not.	
memories.id	true or false
If true, the user ID memory will be programmed, if false, it will not.	
memories.bootflash	true or false
If true, the boot flash (PIC32 only) memory will be programmed, if false, it will not.	
memories.aux	true or false
If true, the auxiliary program memory (dsPIC/PIC24 EP parts only) will be programmed, if false, it will not.	
memories.programmemory.start	a string representing a long value
The value represents the starting program memory address that the debug tool will begin programming.	
memories.programmemory.end	a string representing a long value
The value represents the ending program memory address that the debug tool will end programming.	
programoptions.eraseb4program	true or false
If true, the device will be erased before it is programmed, if false it will not.	
debugoptions.useswbreakpoints	true or false
True indicates that software breakpoints will be used for program address breakpoints, false indicates that hardware breakpoints will be used (does not apply to PICkit 3).	
poweroptions.powerenable	true or false
If true, the debug tool will supply target power at the default voltage for the tool, if false it will not (does not apply to REAL ICE). To set a non-default voltage for the target power, first set the poweroptions.powerenable to true, then set the voltage value where n.n represents the desired voltage: set voltagevalue n.n Example: set poweroptions.powerenable true set voltagevalue 3.3	
system.disableerrormsg	true or false
If true, the system will disable warnings and error messages, if false the system will enable warning and error messages (this is the default).	

© 2012-2013 Microchip Technology Inc.

TABLE 1-7: PROGRAMMING - LIST OF COMMANDS

Program	Programs device memory with the image specified by the file. (Note, if the path or filename has spaces in it, you must use the quotation marks. If there are no spaces in the path of filename, the quotation marks are not needed, as shown below.) `Program executableImageFile`
Upload	Uploads the executable image to MDB memory. `Upload`

TABLE 1-8: RUNNING - LIST OF COMMANDS

Continue	Resume program being debugged, after breakpoint. `Continue`
Halt	Stops the debugged program. `Halt`
Next	Step program, proceeding through subroutine calls. Like the “step” command as long as subroutine calls do not happen; when they do, the call is treated as one instruction. `Next`
Run	Start the debugged program. `Run`
Step	Step program until it reaches a different source line. The step command only enters a function if there si a line number information for the function. `Step`

TABLE 1-9: STACK - LIST OF COMMANDS

Backtrace	Print a backtrace of the entire stack; one line per frame for all frames in the stack. `Backtrace [full] [<n, -n>]` • full - prints the values of local variables • n - prints the innermost n frames • -n - prints the outermost n frames

Programming a Device

The file or hardware tool you need to use for MDB cannot be active or open simultaneously in the MPLAB X IDE, IPE or third party program. Make sure you close or make inactive the file or hardware tool before attempting to use with the MDB.

Note: When programming a device, you must select a device first.

1. Select the device by entering the command:

   `Device [device name]`

   For example: `Device PIC18F66K22`

2. Select the hardware tool. Currently, MDB only supports MPLAB ICD 3, MPLAB PM3, MPLAB REAL ICE, PICkit 3 and Simulator. To verify the supported tools, type:

   `Help Hwtool`

   The MPLAB ICD 3, MPLAB REAL ICE, PICkit 3 and Simulator are for programming and debugging while the MPLAB PM3 is for programming only. To select the hardware tool, type the command:

   `Hwtool [tool name]`

   For example: `Hwtool SIM`

3. If the project was already built, a cof or elf file was generated. To program the device with the cof, elf or hex file, enter the command:

   `Program "[location of the cof or elf or hex file]".`

   For example:

   ```
   Program
   "C:\MDBTestExample\Build\test\preprocess\files\dist\
     \test_IO_Button.cof".
   ```

   If you are using SIM (Simulator) as the hardware tool and the project needs an scl file, it can be set up by using the command:

   `Stim "[location of the scl file]"`

   For more information, use the command Help Stim. You can use Stimulus to set pin injection and/or register injection. An scl file is used to set the condition.

A "Program succeeded" message displays after programming is complete. A verify is automatically performed during a programming sequence.

© 2012-2013 Microchip Technology Inc.

Debugging a Device

You can use the following commands to debug a device.

- MCLR Reset - Refer to the device data sheet for reset information. If an MCLR reset is needed for debugging purposes, enter the command:

  ```
  Reset MCLR
  ```

- Target Device Reset - Refer to the device data sheet for reset information. If you need to reset the target device, first halt the target, then use the command:

  ```
  Help Reset
  ```

  The `Help Reset` command provides information on the usage of the `Reset` Command. It does not actually reset the target.

- Set Breakpoint - There are two ways to set a breakpoint for debugging:
  - Set a breakpoint by source-line-number using the command:

    ```
    Break filename: linenumber
    ```
    For example: `Break main.c:53`

  - Set an absolute address using command:

    ```
    Break *address
    ```
    For example: `Break *0x108`

- Set Watchpoint - To set a watchpoint for debugging:
  - Set a watchpoint by specifying an address and the type of watch using the command:

    ```
    Watch address breakontype
    ```
    For example: `Watch 0xa0007ff0 R`

    or

    ```
    Watch address breakontype[:value] [passcount]
    ```
    For example: `Watch 0xa0007ff0 R:0xf 1`

- Delete Breakpoint - To delete a breakpoint, use the command:

  ```
  Delete [breakpoint number]
  ```

  If no argument is specified in this command, it will delete all breakpoints.

- Run Program - The `Run` command can be used to run the program until it reaches a breakpoint.
- Step Through - To step through the program, use the `Step` command or `Next` command.
- See Variable Value - A `Print [variable]` command can be used to see the value of a variable or an SFR.
- Exit - Use the `Quit` command to exit the MDB.

RUNNING A COMMAND FILE METHOD

> **Note:** Although the MPLAB X IDE can run multiple tools simultaneously, the MDB will not be able to identify each tool.

If programming and debugging needs to be done frequently or multiple times, run the test by running a command file. This will save time over entering the commands repeatedly. Put all the commands in a file on the C drive and run it using this command in Command Prompt, for example:

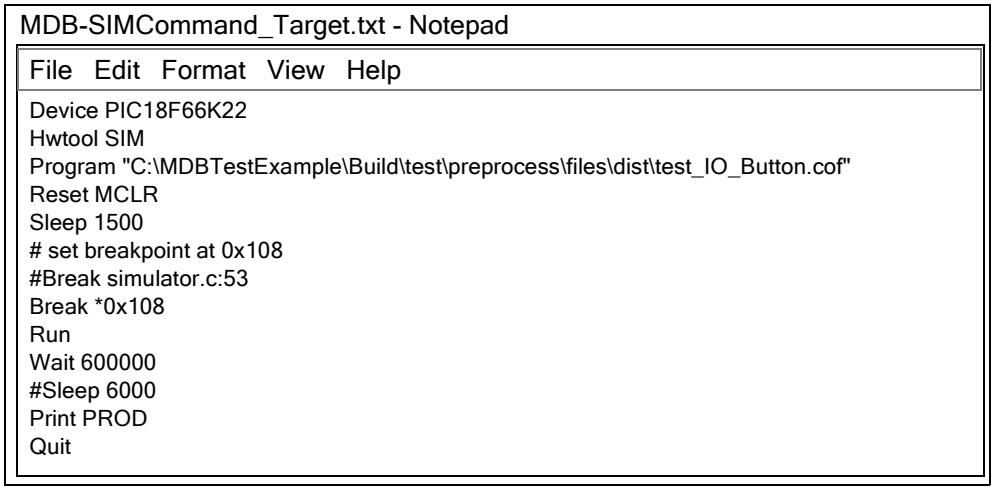
```
C:\Program Files\Microchip\MPLABX\mplab_ide\bin>mdb.bat <commandfile.txt>
```

The following is an example of a command file:

```
C:\MDB-SIMCommand_Target.txt
```

A line starting with # means that it is a comment. A `Sleep` command should be added to make sure the MDB has enough time to finish the previous command before it executes the next command. MDB will run all the commands in the command file sequentially.

FIGURE 1-1: EXAMPLE OF RUNNING A COMMAND FILE



Creating a Printable Log File

By default, the MDB generates xml log files into the MPLAB IDE binary log directory `<MPLAB X installation location>\mplab_ide\bin\log`. Redirecting output to a file is a general option that can be executed from the command prompt and is not specific to the MDB batch file. Redirecting output to a printable text file can be more useful for examining errors than looking at the command prompt window.

To create a printable file, open the `MDB.bat` file and modify it by adding `>>%mplabx_dir%\bin\mdblog.txt` at the end of the batch file. This will make the batch file create the `mdblog.txt` file, which can be printed.

© 2012-2013 Microchip Technology Inc.

Index

A

AutoSelectMemRanges 14

B

Backtrace 15
Break 11, 17
Breakpoints
- Break 11
- Delete 11
- Halt 11
- Watch 11

breakpoints 10

C

Classes of Commands 10
- breakpoints 10
- data 10
- deviceandtool 10
- others 10
- programming 10
- running 10
- stack 10

cof file 16
Command
- Delete 17
- Device 16
- Print 17
- Program 16
- Quit 17
- Reset MCLR 17
- Run 17
- Sleep 18
- Step 17
- Stim 16

Command Prompt 9
Continue 15
Creating a Printable Log File 18

D

Data
- Print 12
- Stim 12
- x 12

data 10
Debugging a Device 17
debugoptions.useswbreakpoints 14
Delete 11
Delete Breakpoint 17
Device 13
Device and Tool
- Device 13
- Hwtool 13

deviceandtool 10
Documentation
- Conventions 5

E

elf file 16
Entering Commands Method 10
Exit 17

H

Halt 11, 15
Header Specification 6
Help 11, 13
Help Reset 17
Hwtool 13

M

MCLR Reset 17
MDB 9
MDB.bat file 18
mdblog.txt file 18
memories.aux 14
memories.bootflash 14
memories.eeprom 14
memories.id 14
memories.programmemory 14
memories.programmemory.end 14
memories.programmemory.start 14
Microchip Debugger 9

N

Next 15

O

Others
- Help 13
- Quit 13
- Set 13
- Sleep 13
- Wait 13

others 10

P

poweroptions.powerenable 14
Print 12
Processor Extension Pak Specification 6
Program 15
Programming
- Program 15
- Upload 15

programming 10
Programming a Device 16
programoptions.eraseb4program 14

© 2012-2013 Microchip Technology Inc.

Q

Quit 13

R

Reading, Recommended 6
Run 15
Run Program 17
Running
    Continue 15
    Halt 15
    Next 15
    Run 15
    Step 15
running 10
Running a Command File Method 18

S

scl file 16
See Variable Value 17
Set 13
    Tool Property Name Options 14
Set Breakpoint 17
Set Watchpoint 17
Simulator 16
Sleep 13
Stack
    Backtrace 15
stack 10
Step 15
Step Through 17
Stim 12
Stimulus 16
system.disableerrormsg 14

T

Target Device Reset 17
Tool Property Name Options 14
    AutoSelectMemRanges 14
    debugoptions.useswbreakpoints 14
    memories.aux 14
    memories.bootflash 14
    memories.eeprom 14
    memories.id 14
    memories.programmemory 14
    memories.programmemory.end 14
    memories.programmemory.start 14
    poweroptions.powerenable 14
    programoptions.eraseb4program 14
    system.disableerrormsg 14
    voltagevalue 14
Transition Socket
    Specification 6

U

Upload 15

V

voltagevalue 14

W

Wait 13
Watch 11

X

x 12
xml log file 18

© 2012-2013 Microchip Technology Inc.

NOTES:

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/12

© 2012-2013 Microchip Technology Inc.